

Foundation of Cryptography, Lecture 4

Pseudorandom Functions

Handout Mode

Iftach Haitner, Tel Aviv University

Tel Aviv University.

April 23, 2013

Section 1

Function Families

function families

- 1 $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$, where $\mathcal{F}_n = \{f: \{0, 1\}^{m(n)} \mapsto \{0, 1\}^{\ell(n)}\}$
- 2 We write $\mathcal{F} = \{\mathcal{F}_n: \{0, 1\}^{m(n)} \mapsto \{0, 1\}^{\ell(n)}\}$
- 3 If $m(n) = \ell(n) = n$, we omit it from the notation
- 4 We identify function with their description
- 5 The rv F_n is uniformly distributed over \mathcal{F}_n

Efficient function families

Definition 1 (efficient function family)

An ensemble of function families $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$ is **efficient**, if:

Samplable. \mathcal{F} is samplable in polynomial-time: there exists a PPT that given 1^n , outputs (the description of) a uniform element in \mathcal{F}_n .

Efficient. There exists a polynomial-time algorithm that given $x \in \{0, 1\}^n$ and (a description of) $f \in \mathcal{F}_n$, outputs $f(x)$.

Random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, let $\Pi_{m,\ell}$ be the family of **all** functions from $\{0, 1\}^m$ to $\{0, 1\}^\ell$.

- It takes $2^m \cdot \ell$ bits to describe an element (i.e., function) of $\Pi_{m,\ell}$.
- We sometimes think of $\pi \in \Pi_{m,\ell}$ as a random string of length $2^m \cdot \ell$.
- Let $\Pi_n = \Pi_{n,n}$

Pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0, 1\}^{m(n)} \mapsto \{0, 1\}^{\ell(n)}\}$ is **pseudorandom**, if

$$\left| \Pr[D^{\mathcal{F}_n}(1^n) = 1] - \Pr[D^{\Pi_{m(n), \ell(n)}}(1^n) = 1] \right| = \text{neg}(n),$$

for any oracle-aided PPT D .

- 1 Suffices to consider $\ell(n) = n$ (why?)
- 2 Easy to construct (with no assumption!) for $m(n) = \log n$ and $\ell \in \text{poly}$
- 3 PRF imply a PRG
- 4 Pseudorandom permutations (PRPs)

Section 2

PRF from OWF

the construction

Construction 4

For $g: \{0, 1\}^n \mapsto \{0, 1\}^{2n}$, let $g_0(s) = g(s)_{1,\dots,n}$ and $g_1(s) = g(s)_{n+1,\dots,2n}$. For $s, x \in \{0, 1\}^*$ define f_s as

$$f_s(x) = g_{x_n}(\dots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathcal{F}_n = \{f_s: s \in \{0, 1\}^n\}$ and $\mathcal{F} = \{\mathcal{F}_n\}$.

If g is an efficient function, then \mathcal{F} is an efficient family.

Theorem 5 (Goldreich-Goldwasser-Micali)

If g is a PRG then \mathcal{F} (defined above) is a PRF.

Corollary 6

OWFs imply PRFs.

Proof Idea

- Easy to prove for inputs of length 2.

Observation: $D = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:

Proof: $D' = (g(U_n^{(0)}), g(U_n^{(1)})) \approx_c U_{4n}$ and $D \approx_c D'$.

- Hence we can handle input of length 2
- Extend to longer inputs?
- We show that an **efficient** sample from the *truth table* of $f \leftarrow \mathcal{F}_n$, is computationally indistinguishable from that of $\pi \leftarrow \Pi_n$.

The Actual Proof

Assume \exists PPT D , $p \in \text{poly}$ and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left| \Pr[D^{F_n}(1^n) = 1] - \Pr[D^{\Pi_n}(1^n) = 1] \right| \geq \frac{1}{p(n)}, \quad (1)$$

for any $n \in \mathcal{I}$, and fix $n \in \mathbb{N}$

Let $t = t(n) \in \text{poly}$ be a bound on the running time of $D(1^n)$. We use D to construct a PPT D' such that

$$\left| \Pr[D'(U_{2n}^t) = 1] - \Pr[D'(g(U_n)^t) = 1] \right| > \frac{1}{np(n)},$$

where $U_{2n}^t = U_{2n}^{(1)}, \dots, U_{2n}^{(t(n))}$ and $g(U_n)^t = g(U_n^{(1)}), \dots, g(U_n^{(t(n))})$.

The Hybrid

Let g and f be as in the definition of \mathcal{F}_n

Definition 7

For $k \in \{0, \dots, n\}$, let $\mathcal{H}_k = \{h_\pi: \{0, 1\}^n \mapsto \{0, 1\}^n: \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_1, \dots, k)}(x_{k+1}, \dots, n)$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0, 1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

- Note that $\mathcal{H}_0 = \mathcal{F}_n$ and $\mathcal{H}_n = \Pi_n$
- Can we emulate \mathcal{H}_k ? We emulate it from **D's point of view**.
- We present efficient "function family" $\mathcal{O}_k = \{O_k^{s^1, \dots, s^t}\}$ s.t.
 - ▶ $D^{O_k^{u_{2n}^t}}(1^n) \equiv D^{H_k}(1^n)$
 - ▶ $D^{O_k^{g(u_n)^t}}(1^n) \equiv D^{H_{k-1}}(1^n)$

for any $k \in [n]$, where H_k is uniformly sampled from \mathcal{H}_k .

Completing the Proof

Let $D'(y)$ return $D_k^{O_k^y}(1^n)$ for k uniformly chosen in $[n]$. Hence

$$\begin{aligned} & \left| \Pr[D'(U_{2n}^t) = 1] - \Pr[D'(g(U_n)^t) = 1] \right| \\ &= \left| \sum_{k=1}^n \frac{1}{n} \cdot \Pr[D_k^{O_k^{U_{2n}^t}}(1^n) = 1] - \sum_{k=1}^n \frac{1}{n} \cdot \Pr[D_k^{O_k^{g(U_n)^t}}(1^n) = 1] \right| \\ &= \frac{1}{n} \left| \sum_{k=1}^n \Pr[D^{H_k}(1^n) = 1] - \sum_{k=1}^n \Pr[D^{H_{k-1}}(1^n) = 1] \right| \\ &= \frac{1}{n} \left| \Pr[D^{H_n}(1^n) = 1] - \Pr[D^{H_0}(1^n) = 1] \right| = \frac{1}{np(n)} \square \end{aligned}$$

The family \mathcal{O}_k

$$\mathcal{O}_k := \{O_k^{s^1, \dots, s^t} : s^1, \dots, s^t \in \{0, 1\}^n \times \{0, 1\}^n\}.$$

Algorithm 8 ($O_k^{s^1, \dots, s^t}$)

On the i 'th query $x^i \in \{0, 1\}^n$:

- 1 If x^ℓ with $x_{1, \dots, k-1}^\ell = x_{1, \dots, k-1}^i$ was **previously** asked, set $z = s_{x_k^i}^\ell$ (where ℓ is the minimal such index).
Otherwise, set $z = s_{x_k^i}^i$ (for $k = 0$ set $z = s_0^1$).
- 2 Return $f_z(x_{k+1, \dots, n}^i)$

- \mathcal{O}_k is **stateful**.

Claim 9

$$D_{O_k^{g(U_n)^t}}(1^n) \equiv D_{O_{k-1}^{u_{2n}^t}}(1^n) \text{ for all } k \in \{1, \dots, n\}.$$

$$D^{O_k^{U_{2n}^t}}(1^n) \equiv D^{H_k}(1^n)$$

Proposition 10

For any $\ell, m \in \mathbb{N}$ and any algorithm A , it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m .

Proof? Does the above trivialize the whole issue of PRF?

Let \tilde{O}_k be the variant of O_k that returns z (and not $f_z(x_{k+1}, \dots, n)$ as in Algorithm 8) and let \tilde{D}_k be the algorithm that implements D using \tilde{O}_k (by computing $f_z(x_{k+1}, \dots, n)$ by itself).

By Proposition 10

$$D^{O_k^{U_{2n}^t}}(1^n) \equiv \tilde{D}_k^{\tilde{O}_k^{U_{2n}^t}}(1^n) \equiv \tilde{D}_k^{\pi_{k,n}}(1^n) \equiv D^{H_k}(1^n) \quad (2)$$

$$D^{O_k^{g(U_n)^t}}(1^n) \equiv D^{H_{k-1}}(1^n)$$

Immediately follows by Claim 9 and Eq 2.

Section 3

PRP from PRF

Pseudorandom permutations

Let $\tilde{\Pi}_n$ be the set of all permutations over $\{0, 1\}^n$.

Definition 11 (pseudorandom permutations)

A **permutation** ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0, 1\}^n \mapsto \{0, 1\}^n\}$ is a pseudorandom permutation, if

$$\left| \Pr[D^{\mathcal{F}_n}(1^n) = 1] - \Pr[D^{\tilde{\Pi}_n}(1^n) = 1] \right| = \text{neg}(n), \quad (3)$$

for any oracle-aided PPT D

- Eq 3 holds for any PRF

PRF to PRP

Definition 12 (LR)

Given $f: \{0, 1\}^n \mapsto \{0, 1\}^n$, the **permutation** $\text{LR}(f): \{0, 1\}^{2n} \mapsto \{0, 1\}^{2n}$ is defined by

$$\text{LR}(f)(\ell, r) = (r, f(r) \oplus \ell).$$

Let $\text{LR}^i(f): \{0, 1\}^{2n} \mapsto \{0, 1\}^{2n}$ be the i 'th iteration of the above operation.

Construction 13

Given a function family $\mathcal{F} = \{\mathcal{F}_n: \{0, 1\}^n \mapsto \{0, 1\}^n\}$, let $\text{LR}^i(\mathcal{F}) = \{\text{LR}^i(\mathcal{F}_n) = \{\text{LR}^i(f): f \in \mathcal{F}_n\}\}$,

$\text{LR}(\mathcal{F})$ is always a permutation family, and is efficient if \mathcal{F} is.

Theorem 14 (Luby-Rackoff)

Assuming that \mathcal{F} is a PRF, then $\text{LR}^3(\mathcal{F})$ is a PRP

Proving Thm 14

It suffices to prove the the following holds for any $n \in \mathbb{N}$ (why?)

Claim 15

$|\Pr[D^{\text{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[D^{\tilde{\Pi}_{2n}}(1^n) = 1]| \leq \frac{4 \cdot q^2}{2^n},$
for **any** q -query algorithm D .

- How would you prove Claim 15?
- Start with non-adaptive D , and show that things only “get wrong” if $\text{LR}(\ell, r) = \text{LR}(\ell', r')$ for two different queries $(\ell, r) \neq (\ell', r')$
- Can you bound the above probability?

Section 4

Applications

General paradigm

Design a scheme assuming that you have random functions, and the **realize** them using PRFs.

Private-key Encryption

Construction 16 (PRF-based encryption)

Given an (efficient) PRF \mathcal{F} , define the encryption scheme $(\text{Gen}, \text{E}, \text{D})$:

Key generation: $\text{Gen}(1^n)$ returns $k \leftarrow \mathcal{F}_n$

Encryption: $\text{E}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption: $\text{D}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

- Advantages over the PRG based scheme?
- Proof of security?